CLAIMS

What is claimed is:

1. A system, comprising:

a first processor having fetch logic and decode logic, the first processor fetches instructions from memory using said fetch logic and decodes said instructions with said decode logic; and

a second processor coupled to said first processor, the second processor fetches an instruction from memory, loads said instruction in the decode logic of the first processor, thereby permitting the first processor to decode said instruction without using the fetch logic.

2. The system of claim 1 wherein said first processor includes a port, the port is coupled to the decode logic and addressable by the second processor.

3. The system of claim 2 wherein said loads said instruction in the decode logic of the first processor comprises writing to a pre-determined address mapped to the port.

4. The system of claim 1 wherein the first processor switches between at least two modes of operation.

5. The system of claim 4 wherein the first processor said fetches instructions from memory using said fetch logic and decodes said instructions with said decode logic in a first mode of said at least two modes of operation.

6. The system of claim 4 wherein the second processor said fetches an instruction from memory, loads said instruction in the decode logic of the first processor, thereby permitting the first processor to decode said instruction without using the fetch logic in a second mode of said at least two modes of operation.

7. A method, comprising:

fetching and decoding instructions in a first processor;

detecting an unsupported instruction that is not executable by the first processor;

executing said unsupported instruction in a second processor; and

providing the first processor with a supported instruction that is executable in the first processor without the first processor fetching said instruction.

8. The method of claim 7 wherein providing the first processor with a supported instruction comprises loading the supported instruction in decode logic of the first processor.

9. The method of claim 7 further comprising detecting patterns of supported and unsupported instructions yet to be executed to determine when to perform said providing the first processor with a supported instruction that is executable in the first processor without the first processor fetching said instruction.

10. The method of claim 9 wherein said patterns comprise an unsupported instruction followed by less than a threshold number of consecutive supported instruction before the next unsupported instruction.

11. A system, comprising:

a first processor having fetch logic and decode logic, the first processor fetches supported instructions from memory using said fetch logic and decodes said supported instructions with said decode logic;

a second processor, the second processor executes unsupported instructions;

means for loading a supported instruction in said decode logic of the first processor so that the first processor decodes but does not fetch the supported instruction;

means for coordinating when said loading a supported instruction in said decode logic of the first processor so that the first processor decodes but does not fetch the supported instruction occurs.

12. The system of claim 11 wherein said means for loading a supported instruction in said decode logic of the first processor so than the first processor decodes but does not fetch the supported instruction comprises coupling the decode logic to a port addressable by the second processor, wherein the second processor fetches the supported instruction and loads said supported instruction in the decode logic of the first processor by accessing said port.

13. The system of claim 12 wherein a switch permits said coupling the decode logic to the port addressable by the second processor.

14. The system of claim 11 wherein said means for coordinating when said loading a supported instruction in said decode logic of the first processor so that the first processor decodes but does not fetch the supported instruction occurs comprises a control program that examines patterns of supported and unsupported instructions yet to be executed and causes the first processor to switch between multiple instruction execution modes according to the patterns, wherein said loading a supported instruction in the decode logic of the first processor so that the first processor decodes but does not fetch the supported instruction is one of said multiple instruction execution modes.

15. The system of claim 14 wherein said control program runs on the second processor.

16. The system of claim 14 wherein said patterns comprise an unsupported instruction followed by less than a threshold number of consecutive supported instruction before the next unsupported instruction.

17. The system of claim 16 wherein said threshold number is three.